Application Report
SPRA487C - September 2001

TMS320C6000 McBSP Interface to SPI ROM

Shaku Anjanaiah
Vassos Soteriou
Digital Signal Processing Solutions

ABSTRACT

The TMS320C6000™ (C6000™) Multichannel Buffered Serial Port (McBSP) is designed to interface to a device that supports synchronous Serial Peripheral Interface (SPI). This document describes the hardware interface between the McBSP and a SPI ROM. The McBSP operates as the master in a user-specified clock stop (CLKSTP) mode in order to communicate with the SPI ROM. The McBSP initialization and control register programming is also discussed. Project collateral discussed in this application report can be downloaded from the following URL: http://www.ti.com/lit/zip/SPRA487.

Contents

List of Figures

List of Tables

1 Design Problem

How do I interface the Serial Peripheral Interface (SPI) ROM to the TMS320C6000?

1.1 Solution

The multichannel buffered serial port (McBSP) in the TMS320C6000 interfaces to a SPI ROM with no glue logic. A SPI system is typically a 4-wire interface comprising serial data in, serial data out, serial clock, and device select. The McBSP provides this 4-wire interface via DR, DX, CLKX, and FSX pins, respectively.

The McBSP supports the SPI interface for a synchronous, full-duplex, variable element length (element length is fixed for a given transfer), master or slave mode back-to-back transmission and reception. This feature is achieved by using the clock-stop (CLKSTP) mode of the McBSP. This document discusses the McBSP interface to an Atmel™ SPI serial CMOS EEPROM, which can only be a slave. The McBSP as a SPI master, generates the required control signals and clocking to the slave.

2 Pin Configuration

For the McBSP to master the interface, you must configure CLKX and FSX pins of the serial port as outputs only. CLKX can be generated either via the C6000 CPU clock or via an external clock source input on the CLKS pin. In SPI mode, a SPI system clock or any other clock source can drive CLKS if present. The clock divide down can be programmed as per application needs.

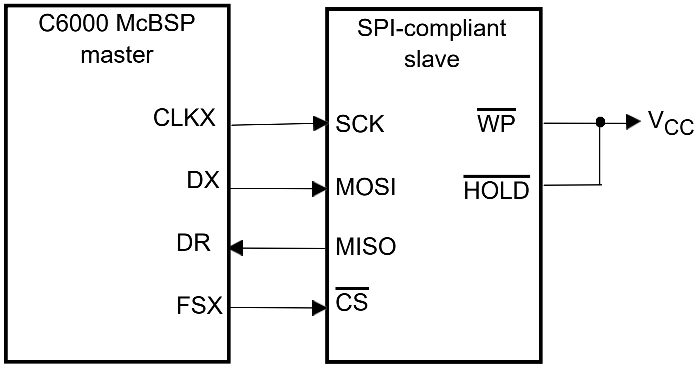

Figure 1. McBSP Master Interface to SPI Slave Device

The signal connectivity shown in Figure 1 is for connecting a Atmel™ SPI serial CMOS EEPROM AT25 series which has a maximum clock rate of 2.1 MHz for Vcc range from 2.7 V to 5.5 V. This slave device is organized as 1k/2k/4k/8k of 8-bit data and only supports SPI modes 0 and 3.

As shown in Figure 1, the SPI Mode supports simultaneous transmission and reception utilizing signals that correspond to the transmitter. The McBSP can simultaneously receive data since the CLKR and FSR signals are driven by CLKX and FSX signals (respectively) internally. As a good practice, CLKR and FSR should be programmed as inputs.

3 McBSP Initialization

The various McBSP control registers shown in Figure 2 through Figure 6 have to be initialized for the SPI operation. The serial port initialization procedure for SPI mode is as follows:

1. If McBSP is not in reset state, set $\overline{\text{XRST}}$ = $\overline{\text{RRST}}$ = 0 in SPCR.
2. Program the McBSP configuration registers XCR, RCR, SRGR, PCR, and SPCR for all parameters as required. Write the desired value into the CLKSTP bit-fields in the SPCR. Figure 7 shows the various CLKSTP modes that are supported by the McBSP.
3. Set $\overline{\text{GRST}}$ = 1 in SPCR to get the sample rate generator out of reset.
4. Wait two bit clocks for the McBSP to reinitialize.
5. Either a, b or c should be followed.
   a. This step should be performed if the CPU is used to service the McBSP. Set $\overline{\text{XRST}}$ = $\overline{\text{RRST}}$ = 1 to enable the serial port. Note that the value written to the SPCR at this time should have only the reset bits changed to 1 and the remaining bit-fields should have the same value as in Step 2 above.
   b. If the DMA is used to perform data transfers, it should first be initialized with the appropriate read/write syncs, src/dst addresses, and their update modes, transfer complete interrupt, and any other feature suitable for the application. Lastly, set the START bit. The DMA is now in the START state and waits for the synchronization events to occur. Then, pull the McBSP out of reset. For details on DMA initialization for servicing the McBSP, refer to *TMS320C6000 McBSP Initialization* (SPRA488) and *TMS320C6000 DMA Applications* (SPRA529).
   c. If the enhanced DMA (EDMA) is used to perform data transfers, the channels associated to the McBSP transmit and receive synchronization events should first be configured with the appropriate priority levels, element size, src/dst addresses, address update modes, transfer complete code, transfer complete interrupt enable, source and destination dimensions, and any other feature suitable for the application in the PaRAM parameter fields. The events are latched in the event register (ER), even in the events are disabled. Enabling the corresponding event bit in the event enable register (EER) starts the data transfer by setting this bit to a ‘1’. Then, pull the McBSP out of reset. For details on EDMA initialization for servicing the McBSP, refer to *TMS320C6000 McBSP Initialization* (SPRA488) and *TMS320C6000 Enhanced DMA: Example Applications* (SPRA636).

31	30 24	23 21	20 19	18	17 16
RPHASE	RFRLEN2	RWDLEN2	RCOMPAND	RFIG	RDATDLY
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14 8	7 5	4	3 0
RPHASE2†	RFRLEN1	RWDLEN1	RWDREVRS†	reserved
R/W-0	R/W-0	R/W-0	R/W-0	R-0

Legend: R = Read only; R/W = Read/Write
† Available only on C621x/C671x and C64x devices.

Figure 2. Receive Control Register (RCR for SPI Master)

31	30 24	23 21	20 19	18	17 16
XPHASE	XFRLEN2	XWDLEN2	XCOMPAND	XFIG	XDATDLY
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14 8	7 5	4	3 0
XPHASE2†	XFRLEN1	XWDLEN1	XWDREVRS†	reserved
R/W-0	R/W-0	R/W-0	R/W-0	R-0

Legend: R = Read only; R/W = Read/Write
† Available only on C621x/C671x and C64x devices.

Figure 3. Transmit Control Register (XCR for SPI Master)

31	30	29	28	27 16
GSYNC	CLKSP	CLKSM	FSGM	FPER
R/W-0	R/W-0	R/W-1	R/W-0	R/W-0

15 8	7 0
FWID	CLKGDV
R/W-0	R/W-1

Legend: R/W = Read/Write

Figure 4. Sample Rate Generator Register (SRGR for SPI Master)

31 16
Reserved
R-0

15 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsvd	XIOEN	RIOEN	FSXM	FSRM	CLKXM	CLKRM	Rsvd	CLKS_STAT	DX_STAT	DR_STAT	FSXP	FSRP	CLKXP	CLKRP
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read only; R/W = Read/Write

Figure 5. Pin Control Register (PCR for SPI Master)

31 26	25	24	23	22	21 20	19	18	17	16
Reserved	FREE+	SOFT+	$\overline{\text{FRST}}$	GRST	XINTM	XSYNCERR†	XEMPTY	XRDY	XRST
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R/W-0

15	14 13	12 11	10 8	7	6	5 4	3	2	1	0
DLB	RJUST	CLKSTP	Reserved	DXENA‡	Reserved	RINTM	RSYNCERR†	RFULL	RRDY	$\overline{\text{RRST}}$
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R/W-0	R-0	R-0	R/W-0

Legend: R = Read only; R/W = Read/Write

† Writing a 1 to XSYNCERR or RSYNCERR will set the error condition when the transmiter or receiver ($\overline{\text{XRST}}$=1 or $\overline{\text{RRST}}$=1), respectively, are enabled. Thus, it is used mainly for testing purposes or if this operation is desired.

‡ Available in the C621x/C671x/C64x only.

R/W–x = Read/write–reset value

Figure 6. Serial Port Control Register (SPCR for SPI Master)

Table 1. McBSP Register Values for 200 MHz CPU Clock

Register	Value in Hex	Description
RCR	0x000100A0	Single phase, one 32-bit element per frame, one bit-clock delay
XCR	0x000100A0	Single phase, one 32-bit element per frame, one bit-clock delay
SRGR	0x2000005F	• Serial clock CLKX generated by internal clock (CLKSM = 1). The internal clock source is the CPU clock for C620x/C670x, the CPU/2 clock for C621x/C671x and the CPU/4 clock for C64x. • Frame sync FSX generated due to DXR-to-XSR transfer (FSGM = 0) • Clock divide down is 95 for 200 MHz clock to generate 2.08 MHz shift clock (CLKGDV = 0x5F). CLKGCV should be adjusted accordingly for other internal clock source rate.
PCR	0x00000A0C	• FSX is an active-low (FSXP = 1) output (FSXM = 1) • FSR is an active-low (FSRP = 1) input (FSRM = 0) • CLKX is an output (CLKXM = 1) and starts with a rising edge (CLKXP = 0)
SPCR	0x00001800	CLKSTP = 11b. Since CLKXP=0, this refers to data transmitted on rising edge and received on falling edge of CLKX by the master. This parameter can be changed as per application needs.

The example code initializes McBSP0 in the correct order for SPI-mode communication between the McBSP and a SPI serial EEPROM. The zip file of code is available with this application report.

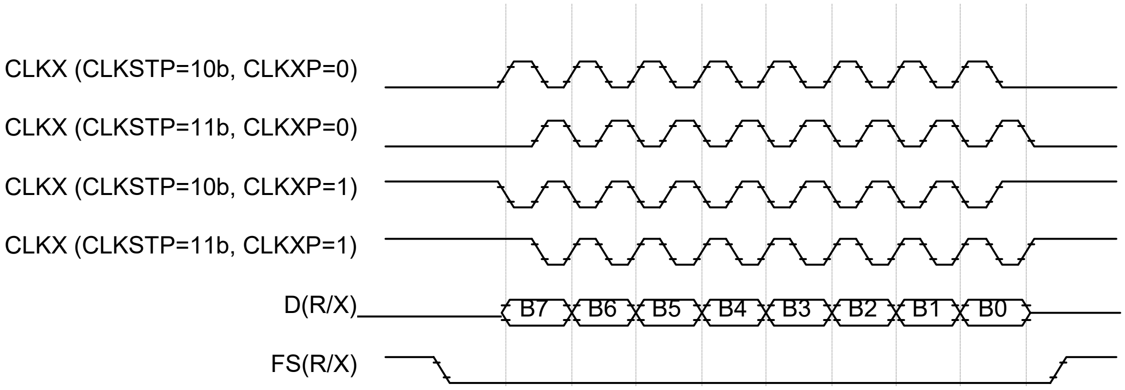
CLKX (CLKSTP=10b, CLKXP=0)
CLKX (CLKSTP=11b, CLKXP=0)
CLKX (CLKSTP=10b, CLKXP=1)
CLKX (CLKSTP=11b, CLKXP=1)
D(R/X)
B7
B6
B5
B4
B3
B2
B1
B0
FS(R/X)

Figure 7. Clock Stop Mode Options

4 Timing Analysis

SPI mode (0,0) of the SPI ROM corresponds to the McBSP SPI mode with CLKSTP = 11b and CLKXP = 0. The master (McBSP) shifts data out on the falling edge of CLKX and the slave (SPI ROM) samples the receive data on the rising edge of CLKX. The slave transmits/shifts data out on the falling edge of CLKX and the master samples the receive data on the rising edge of CLKX.

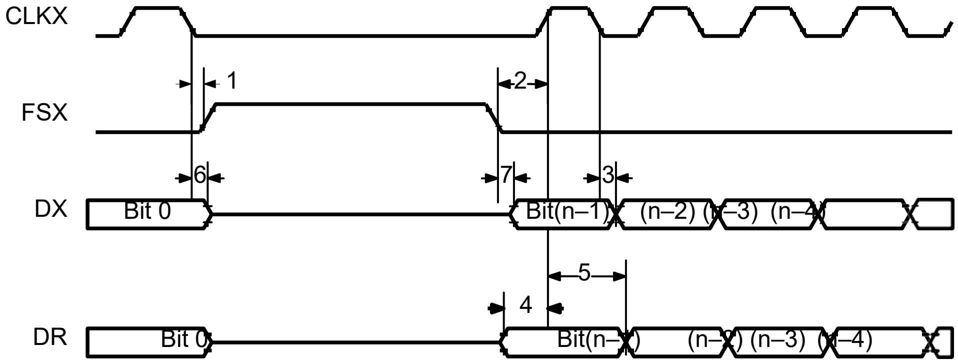

Figure 8. C6000 Timing, CLKSTP = 11b, CLKXP = 0

The timing diagram for CLKSTP=11b, and CLKXP=0 is shown in Figure 8. The corresponding values for the timing requirements and switching characteristics for a 2.1 MHz operation are shown in Table 2. The values are derived from the formula/numbers available in the TMS320C6201 datasheet. The AC timing on different C6000 devices may also differ. Please refer to the specific device data sheet and replace the values in Table 2 for timing analysis of a particular device.

Table 2. Timing Numbers for McBSP as SPI Master

Switching Characteristics

NO.	Parameter		Min	Max	UNIT
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high	L–2 = 238	L+3 =243	ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low	T–2 = 478	T+3 = 483	ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	–2	4	ns
6	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	–2	4	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	H–2 = 238	H+4 =244	ns

Timing Requirements

NO.			Min	Max	Unit
4	$t_{su(DRV-CKXL)}$	Setup time, DR valid before CLKX low	12		ns
5	$t_{h(CKXL-DRV)}$	Hold time, DR valid after CLKX low	4		ns

NOTE: The following is true for the above calculations:

1. Since CLKGDV = 95, CLKX derived from CPU clock will have a 50% duty cycle and therefore H = L.
2. Period of CLKX, T = (1 + CLKGDV) * P where P = 5 ns for a 200 MHz CPU clock. Hence, T = 480 ns.

As shown in Table 3, the timing numbers of the McBSP match with that of the SPI ROM with sufficient timing margins. Note that the SPI ROM timings correspond to the 2.7 V–5.5 V range of devices. Therefore a voltage translation buffer (for example, SN54LVT16373) will have to be used between the McBSP and the SPI ROM. Use of buffers will still meet the necessary timing requirements/margins.

Table 3. Timing Analysis for SPI Master and Slave

SPI ROM Switching Characteristics	
Parameter	**Value**
$t_{wl(min)} - t_{v(max)}$;;where t_{wl} = L = 240 – 200	40 ns
$t_{wh(min)} + t_{ho(min)}$;; where t_{wh} = H = 240 + 0	240 ns
SPI ROM Timing Requirements	
	Value
$t_{su(min)}$ Setup time, data in	50 ns
$t_{h(min)}$ Hold time, data in	50 ns
$t_{css(min)}$ Setup time, $\overline{CS}$	250 ns
C6201 Switching Characteristics	
Parameter	**Value**
$t_{d(FXL-CKXL)min} - t_{d(FXL-DXV)max}$ = 478 – 245	234 ns
H + $t_{d(CKXH-DXV)min}$ = 240 – 2	238 ns
$t_{d(FXL-CKXL)min}$	478 ns
C6201 Timing Requirements	
	Value
$t_{su(DRV-CKXL)min}$ Setup time, data in	12 ns
$t_{h(CKXL-DRV)min}$ Hold time, data in	4 ns

For applications where the McBSP is used as a SPI slave, please ensure that the internal clock, CLKG runs at least eight times that of the master clock. Typically, programming CLKGDV = 1 and using CPU clock (CLKSM = 1) (when McBSP is a SPI slave) should suffice since SPI clocks are very slow.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265